Technische Universität Karl-Marx-Stadt
Sektion: Fertigungsprozeß und Fertigungsmittel
WB:

Grundlagen der Mikrorechnerprogrammierung

Befehlsliste K 1520

Befehlsliste U 880

Flagbeeinflussung, Taktzyklenangabe, Funktionssymbolik

Erläuterung der verwendeten Symbolik:

r, r_1, r_2,	8-Bit-Register:	A, B, C, D, E, H, L
dd	16-Bit-Register,	BC, DE, HL, SP
qq	16-Bit-Register:	BC, DE, HL, AF
rr	16-Bit-Register:	BC, DE, IX, SP
ss	16-Bit-Register:	BC, DE, IY, SP
cc	Bedingungskode:	NZ ungleich Null Z gleich Null NC kein Übertrag C Übertrag PO ungerade Parität bzw. kein Überlauf PE gerade Parität bzw. Überlauf P positiv (Bit 7 = 0) N negativ (Bit 7 = 1)

H, L als Index höherwertiger bzw. niederwertiger Teil eines 16-Bit-Registers

n	8-Bit-Wert:	0...255
nn	16-Bit-Wert:	0...65535
d	8-Bit-Abstandswert bei indizierter Adressierung (IX, IY) in Zweierkomplementdarstellung: - 128 ... + 127	
e	8-Bit-Abstandswert bei relativen Sprüngen in Zweierkomplementdarstellung: - 126 ... + 129	
Achtung:	Im Operationskode ist e-2 einzusetzen, da PC vor der Addition mit e bereits intern um 2 erhöht wird.	
p	Kurzrufadresse (ØØH, Ø8H, 1ØH, 18H, 2ØH, 28H, 3ØH, 38H)	
s_b	Bit eines Registers oder einer Speicherstelle: b = Ø, 1, 2, 3, 4, 5, 6, 7	
()	Ausdruck zwischen den Klammern stellt einen Zeiger (Adresse) auf eine Speicherstelle oder E/A-Port dar Anm.: Statt (HL) kann auch M stehen	

Erläuterungen zum Flagsatz

Flag	Bedeutung	
CY	Übertrag:	CY=1, wenn Operation einen Übertrag erzeugt bzw. bei Verschiebeop. eine 1 herausgeschoben wurde
Z	Null:	Z=1, wenn Ergebnis einer Operation gleich Null, bei Blocksuchbefehlen Übereinstimmung gefunden
S	Vorzeichen:	S=1, wenn beim Ergebnis einer Operation Bit7=1 / Parität (Anzahl der 1-Bits)
P/V	Überlauf:	V=1, wenn bei arithmetischen Operationen in der Zweierkomplementdarstellung der zulässige Bereich -128...+127 verlassen wurde
	bei Blocktransfer- und Blocksuchbefehlen:	P/V=1, wenn BC noch nicht Null ist
	bei LD A,I/LD A,R:	P/V:=IFF2 (Unterbrechungserlaubnis-Flip-flop)
H	Halbbyte-Übertrag:	H=1, wenn Übertrag vom niederwertigen in das höherwertige Halbbyte auftritt
N	Addition/Subtraktion:	N=1, wenn vorangegangene Operation eine Subtraktion war

Symbolik der Flagbeeinflussung

↕	Flag wird entsprechend dem Ergebnis der Operation gestellt
.	Flag wird durch die Operation nicht beeinflußt
0	Flag wird durch die Operation gelöscht
1	Flag wird durch die Operation gesetzt
x	Flag unbestimmt

Besonderheiten:

a) P/V-Flag:=0, wenn BC Null erreicht hat, sonst P/V:=1

b) Z-Flag:=1, wenn A= (HL), sonst P/V:=1

c) Z-Flag:=1, wenn B Null erreicht hat, sonst Z:=0

Aufbau des Flag-Bytes:

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	X	H	X	P/V	N	CY

X - D3 und D5 enthalten keine Flagbedeutung

nur bei Blockbefehlen

:= wird zu

( ) Inhalt von Speicherzelle

Assembler-sprache	symbolische Operation	Flags CY Z P/V S N H	Op.-Kode 76 543 210	Takte	Bemerkungen
8 - Bit - Ladebefehle					
LD r_1, r_2	r_1:=r_2	. . . . .	01 r_1 r_2	4	r_1,r_2: Register
LD r,n	r:=n	. . . . .	00 r 110 - n -	7	r: 000 B, 001 C, 010 D, 011 E, 100 H, 101 L, 111 A
LD r, (HL)	r:=(HL)	. . . . .	01 r 110	7	
LD r,(IX+d)	r:=(IX+d)	. . . . .	11 011 101 01 r 110 - d -	19	
LD r,(IY+d)	r:=(IY+d)	. . . . .	11 111 101 01 r 110 - d -	19	
LD (HL),r	(HL):=r	. . . . .	01 110 r	7	
LD (IX+d),r	(IX+d):=r	. . . . .	11 011 101 01 110 r - d -	19	
LD (IY+d),r	(IY+d):=r	. . . . .	11 111 101 01 110 r - d -	19	
LD (HL),n	(HL):=n	. . . . .	00 110 110 - n -	10	
LD (IX+d),n	(IX+d):=n	. . . . .	11 011 101 00 110 110 - d - - n -	19	
LD (IY+d),n	(IY+d):=n	. . . . .	11 111 101 00 110 110 - d - - n -	19	
LD A, (BC)	A:=(BC)	. . . . .	00 001 010	7	
LD A, (DE)	A:=(DE)	. . . . .	00 011 010	7	
LD A, (nn)	A:=(nn)	. . . . .	00 111 010 - n - - n -	13	
LD (BC),A	(BC):=A	. . . . .	00 000 010	7	
LD (DE),A	(DE):=A	. . . . .	00 010 010	7	
LD (nn),A	(nn):=A	. . . . .	00 110 010 - n - - n -	13	
LD A,I	A:=I	. ↕ IFF ↕ 0 0	11 101 101 01 010 111	9	
LD A,R	A:=R	. ↕ IFF ↕ 0 0	11 101 101 01 011 111	9	
LD I,A	I:=A	. . . . .	11 101 101 01 000 111	9	

LD R,A

Assembler-sprache	symbolische Operation	Flags CY	Z	P/V	S	N	H	Op.-Kode 76 543 210	Takte	Bemerkungen
16-Bit-Ladebefehle										
LD dd, nn	dd:=nn	.	.	.	.	.	.	00 dd0 001 - n - - n -	10	dd Reg. 00 BC 01 DE 10 HL 11 SP
LD IX,nn	IX:=nn	.	.	.	.	.	.	11 011 101 00 100 001 - n - - n -	14	
LD IY,nn	IY:=nn	.	.	.	.	.	.	11 111 101 00 100 001 - n - - n -	14	
LD HL,(nn)	H:=(nn+1) L:=(nn)	.	.	.	.	.	.	00 101 010 - n - - n -	16	
LD dd,(nn)	dd_H:=(nn+1) dd_L:=(nn)	.	.	.	.	.	.	11 101 101 01 dd1 011 - n - - n -	20	
LD IX,(nn)	IX_H:=(nn+1) IX_L:=(nn)	.	.	.	.	.	.	11 011 101 00 101 010 - n - - n -	20	
LD IY,(nn)	IY_H:=(nn+1) IY_L:=(nn)	.	.	.	.	.	.	11 111 101 00 101 010 - n - - n -	20	
LD (nn),HL	(nn+1):=H (nn):=L	.	.	.	.	.	.	00 100 010 - n - - n -	16	
LD (nn),dd	(nn+1):=dd_H (nn):=dd_L	.	.	.	.	.	.	11 101 101 01 dd0 011 - n - - n -	20	
LD (nn),IX	(nn+1):=IX_H (nn):=IX_L	.	.	.	.	.	.	11 011 101 00 100 010 - n - - n -	20	
LD (nn),IY	(nn+1):=IY_H (nn):=IY_L	.	.	.	.	.	.	11 111 101 00 100 010 - n - - n -	20	
LD SP, HL	SP:=HL	.	.	.	.	.	.	11 111 001	6	
LD SP, IX	SP:=IX	.	.	.	.	.	.	11 011 101 11 111 001	10	
LD SP, IY	SP:=IY	.	.	.	.	.	.	11 111 101 11 111 001	10	

Assembler-sprache	symbolische Operation	Flags CY	Z	P/V	S	N	H	Op.-Kode 76 543 210	Takte	Bemerkungen
PUSH qq	$(SP-2):=qq_L$ $(SP-1):=qq_H$ $SP:=SP-2$	.	.	.	.	.	.	11 qq0 101	11	qq | Reg. 00 | BC 01 | DE 10 | HL 11 | AF
PUSH IX	$(SP-2):=IX_L$ $(SP-1):=IX_H$ $SP:=SP-2$	.	.	.	.	.	.	11 011 101 11 100 101	15	
PUSH IY	$(SP-2):=IY_L$ $(SP-1):=IY_H$ $SP:=SP-2$	.	.	.	.	.	.	11 111 101 11 100 101	15	
POP_{qq}	$qq_L:=(SP)$ $qq_H:=(SP+1)$ $SP:=SP+2$	.	.	.	.	.	.	11 qq0 001	10	
POP IX	$IX_L:=(SP)$ $IX_H:=(SP+1)$ $SP:=SP+2$	.	.	.	.	.	.	11 011 101 11 100 001	14	
POP IY	$IY_L:=(SP$ $IY_H:=(SP+1)$ $SP:=SP+2$	.	.	.	.	.	.	11 111 101 11 100 001	14	
Bit testen, setzen, rücksetzen										
BIT b,r	$Z:=\overline{r_b}$	.	↕		x	x	0	1	11 001 011 01 b r	8	r | Reg. 000 | B 001 | C 010 | D 011 | E 100 | H 101 | L 111 | A
BIT b,(HL)	$Z:=\overline{(HL)_b}$	.	↕		x	x	0	1	11 001 011 01 b 110	12	
BIT b,(IX+d)	$Z:=\overline{(IX+d)_b}$	.	↕		x	x	0	1	11 011 101 11 001 011 - d - 01 b 110	20	
BIT b,(IY+d)	$Z:=\overline{(IY+d)_b}$		↕		x	x	0	1	11 111 101 11 001 011 - d - 01 b 110	20	b | Bit 000 | 0 001 | 1 010 | 2 011 | 3 100 | 4 101 | 5 110 | 6 111 | 7
SET b, r	$r_b:=1$	.	.	.	.	.	.	.	11 001 011 11 b r	8	
SET b,(HL)	$(HL)_b:=1$	.	.	.	.	.	.	.	11 001 011 11 b 110	15	
SET b,(IX+d)	$(IX+d)_b:=1$	.	.	.	.	.	.	.	11 011 101 11 001 011 - d - 11 b 110	23	
SET b,(IY+d)	$(IY+d)_b:=1$	.	.	.	.	.	.	.	11 111 101 11 001 011 - d - 11 b 110	23	
RES b,s	$S_b:=0$ s=r,(HL) (IX+d),(IY+d)								10	Zur Bildg. d.Op.-Kodes 11 in SET durch 10 ersetzen,

Assembler-sprache	symbolische Operation	Flags CY	Z	P/V	S	N	H	Op.-Kode 76 543 210	Takte	Bemerkungen
Austausch-, Blocktransfer- und Blocksuchbefehle										
EX. DE, HL	DE↔HL	.	.	.	.	.	.	11 101 011	4	
EXAF	AF↔AF'	.	.	.	.	.	.	00 001 000	4	
EXX	BC↔BC' DE↔DE' HL↔HL'	.	.	.	.	.	.	11 011 001	4	Vertauschung der Reg.-sätze
EX (SP),HL	H↔(SP+1) L↔(SP)	.	.	.	.	.	.	11 100 011	19	
EX (SP),IX	IXH↔(SP+1) IXL↔(SP)	.	.	.	.	.	.	11 011 101 11 100 011	23	
EX (SP),IY	IYH↔(SP+1) IYL↔(SP)	.	.	.	.	.	.	11 111 101 11 100 011	23	
LDI	(DE):=(HL) DE:=DE+1 HL:=HL+1 BC:=BC-1	.	x	↕a	x	0	0	11 101 101 10 100 000	16	
LDIR	(DE):=(HL) DE:=DE+1 HL:=HL+1 BC:=BC-1 Wiederholung bis BC=0	.	x	0	x	0	0	11 101 101 10 110 000	21 16	wenn BC≠0 wenn BC=0
LDD	(DE):=(HL) DE:=DE-1 HL:=HL-1 BC:=BC-1	.	x	↕a	x	0	0	11 101 101 10 101 000	16	
LDDR	(DE):=(HL) DE:=DE-1 HL:=HL-1 BC:=BC-1 Wiederholung bis BC=0	.	x	0	x	0	0	11 101 101 10 111 000	21 16	wenn BC≠0 wenn BC=0
CPI	A- (HL) HL:=HL+1 BC:=BC-1	.	↕b	↕a	↕	1	↕	11 101 101 10 100 001	16	
CPIR	A - (HL) HL:=HL+1 BC:=BC-1 Wiederholung bis BC=0 oder A =(HL)	.	↕b	↕a	↕	1	↕	11 101 101 10 110 001	21 16	wenn BC≠0 wenn BC=0
CPD	A - (HL) HL:=HL-1 BC:=BC-1	.	↕b	↕a	↕	1	↕	11 101 101 10 101 001	16	
CPDR	A - (HL) HL:=HL-1 BC:=BC-1 Wiederholung bis BC=0 oder =(HL)	.	↕b	↕a	↕	1	↕	11 101 101 10 111 001	21 16	wenn BC≠0 wenn BC=0

Assembler-sprache	symbolische Operation	Flags CY	Z	P/V	S	N	H	Op.-Kode 76 543 210	Takte	Bemerkungen
8-Bit-Arithmetik und Logik										
ADD r	A:=A+r	↕	↕	V	↕	0	↕	10 [000] r	4	r | Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
ADD n	A:= A+n	↕	↕	V	↕	0	↕	11 [000] 110 - n -	7	
ADD (HL)	A:=A+(HL)	↕	↕	V	↕	0	↕	10 [000] 110	7	
ADD (IX+d)	A:=A+(IX+d)	↕	↕	V	↕	0	↕	11 011 101 10 [000] 110 - d -	19	
ADD (IY+d)	A:=A+(IY+d)	↕	↕	V	↕	0	↕	11 111 101 10 [000] 110 - d -	19	
ADC s	A:=A+s+CY	↕	↕	V	↕	0	↕	[001]		s ist r,n (HL)(IX+d) oder (IY+d) wie bei ADD.
SUB s	A:=A-s	↕	↕	V	↕	1	↕	[010]		
SBC s	A:=A-s-CY	↕	↕	V	↕	1	↕	[011]		
AND s	A:=A∧s	0	↕	P	↕	0	1	[100]		umrandete Bits ersetzen 000 in ADD
OR s	A:=A∨s	0	↕	P	↕	0	0	[110]		
XOR s	A:=A⊻s	0	↕	P	↕	0	0	[101]		
CMP s	A - s	↕	↕	V	↕	1	↕	[111]		
INC r	r:=r+1	•	↕	V	↕	0	↕	00 r [100]	4	
INC (HL)	(HL):=(HL)+1	•	↕	V	↕	0	↕	00 110 [100]	11	
INC (IX+d)	(IX+d) :=(IX+d)+1	•	↕	V	↕	0	↕	11 011 101 00 110 [100] - d -	23	
INC (IY+d)	(IY+d) :=(IY+d)+1	•	↕	V	↕	0	↕	11 111 101 00 110 [100] - d -	23	
DEC t	t:=t-1	•	↕	V	↕	1	↕	[101]		t ist r, (HL)(IX+d) oder(IX+d) wie beiINC. 100 durch 101 im Op.-kode ersetzen

SCF

Assembler-sprache	Symbolische Operation	Flags CY	Z	P/V	C	N	H	Op.-Kode 76 543 210	Takte	Bemerkungen
Allgemeine Arithmetik und ZVE-Steuerbefehle										
DAA	Korrektur von A nach Add./Sub.gepackter BCD-Zahlen	↕	↕	P	↕	·	↕	00 100 111	4	
CPL	A:=$\overline{A}$	·	·	·	·	1	1	00 101 111	4	bitweise Negation Einerkompl.
NEG	A:=$\overline{A}$+1	↕	↕	V	↕	1	↕	11 101 101 01 000 100	8	Zweierkomplement
CCF	CY:=$\overline{CY}$	↕	·	·	·	0	X	00 111 111	4	Kompl.d. Übertr.Flags
NOP	keine Op.	·	·	·	·	·	·	00 000 000	4	
HALT	ZVE in HALT	·	·	·	·	·	·	01 110 110	4	
DI	IFF:=0	·	·	·	·	·	·	11 110 011	4	Int sperren
EI	IFF:=1	·	·	·	·	·	·	11 111 011	4	Int erlaub.
IM0	Setzen Int.-Mode 0	·	·	·	·	·	·	11 101 101 01 000 110	8	
IM1	Setzen Int.-Mode 1	·	·	·	·	·	·	11 101 101 01 010 110	8	
IM2	Setzen Int.-Mode 2	·	·	·	·	·	·	11 101 101 01 011 110	8	
16-Bit-Arithmetik										
ADD HL,dd	HL:=HL+dd	↕	·	·	·	0	x	00 dd1 001	11	
ADC HL,dd	HL:=HL+dd+CY	↕	↕	V	↕	0	x	11 101 101 01 dd1 010	15	
SBC HL,dd	HL:=HL-dd-CY	↕	↕	V	↕	1	x	11 101 101 01 dd0 010	15	
ADD IX,rr	IX:=IX+rr	↕	·	·	·	0	x	11 011 101 00 pp1 001	15	
ADD IY,ss	IY:= IY+ss	↕	·	·	·	0	x	11 111 101 00 ss1 001	15	
INC dd	dd:=dd+1	·	·	·	·	·	·	00 dd0 011	6	
INC IX	IX:=IX+1	·	·	·	·	·	·	11 011 101 00 100 011	10	
INC IY	IY:=IY+1	·	·	·	·	·	·	11 111 101 00 100 011	10	
DEC dd	dd:=dd-1	·	·	·	·	·	·	00 dd1 011	6	
DEC IX	IX:=IX-1	·	·	·	·	·	·	11 011 101 00 101 011	10	
DEC IY	IY:=IY-1	·	·	·	·	·	·	11 111 101 00 101 011	10	

dd	Reg.
00	BC
01	DE
10	HL
11	SP

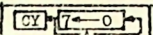
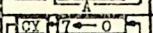
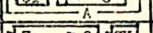
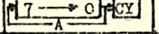
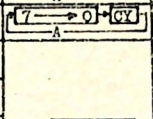
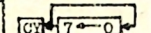
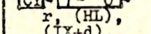
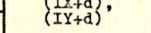
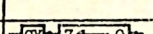
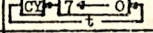
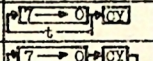
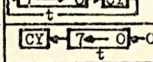
rr	Reg.
00	BC
01	DE
10	IX
11	SP

ss	Reg.
00	BC
01	DE
10	IY
11	SP

Befehl auf Datenbus ≙ RST 38

Portadressbildg

NMI ≙ RST 66

Assembler-sprache	Symbolische Operation	CY	Z	P/V	S	N	H	Op.-Kode 76 543 210	Takte	Bemerkungen
Verschiebe- und Rotationsbefehle										
RLCA		↕	·	·	·	0	0	00 000 111	4	
RLA		↕	·	·	·	0	0	00 010 111	4	
RRCA		↕	·	·	·	0	0	00 001 111	4	
RRA		↕	·	·	·	0	0	00 011 111	4	
RLC r		↕	↕	P	↕	0	0	11 001 011 00 000 r	8	
RLC (HL)		↕	↕	P	↕	0	0	11 001 011 00 000 110	15	r: 000 B, 001 C, 010 D, 011 E, 100 H, 101 L, 111 A
RLC (IX+d)		↕	↕	P	↕	0	0	11 011 101 11 001 011 d 00 000 110	23	
RLC (IY+d)		↕	↕	P	↕	0	0	11 111 101 11 001 011 d 00 000 110	23	
RL t		↕	↕	P	↕	0	0	010		t ist r, (HL), (IX+d), (IY+d) wie bei RLC umrandete Bits ersetzen 000 in RLC
RRC t		↕	↕	P	↕	0	0	001		
RR t		↕	↕	P	↕	0	0	011		
SLA t		↕	↕	P	↕	0	0	100		
SRA t		↕	↕	P	↕	0	0	101		
SRL t		↕	↕	P	↕	0	0	111		
RLD		·	↕	P	↕	0	0	11 101 101 01 101 111	18	Verschiebung um eine Dezimalstelle zw. A u. (HL) obere Hälfte von unbeeinflußt 1)
RRD		·	↕	P	↕	0	0	11 101 101 01 100 111	18	

1) Links u. Rechtsrotation der Tetraden.

Assembler-sprache	Symbolische Operation	Flags CY	Z	P/V	S	N	H	Op.-Kode 76 543 210	Takte	Bemerkungen
Sprünge, Unterprogrammaufrufe, Rücksprünge										
JMP nn	PC:=nn	.	.	.	.	.	.	11 000 011 - n - - n -	10	cc Beding. 000 NZ 001 Z 010 NC 011 C 100 PO 101 PE 110 P 111 M
JPcc nn	wenn cc erfüllt dann PC:=nn	.	.	.	.	.	.	11 cc 010 - n - - n -	10	
JR e	PC:=PC+e	.	.	.	.	.	.	00 011 000 - e-2 -	12	
JRC e	wenn CY=1, dann PC:=PC+e	.	.	.	.	.	.	00 111 000 - e-2 -	7 12	
JRNC e	wenn CY=0, dann PC:=PC+e	.	.	.	.	.	.	00 110 000 - e-2 -	7 12	
JRZ e	wenn Z=1, dann PC:=PC+e	.	.	.	.	.	.	00 101 000 - e-2 -	7 12	
JRNZ e	wenn Z=0 dann PC:=PC+e	.	.	.	.	.	.	00 100 000 - e-2 -	7 12	
JMP (HL)	PC:=HL	.	.	.	.	.	.	11 101 001	4	
JMP (IX)	PC:=IX	.	.	.	.	.	.	11 011 101 11 101 001	8	
JMP (IY)	PC:=IY	.	.	.	.	.	.	11 111 101 11 101 001	8	
DJNZ e	B:=B-1 wenn B≠0, dann PC:=PC+e	.	.	.	.	.	.	00 010 000 - e-2 -	8 13	wenn B=0 wenn B≠0
CALL nn	(SP-1):=PC_H (SP-2):=PC_L PC:= nn SP:=SP-2	.	.	.	.	.	.	11 001 101 - n - - n -	17	
CAcc nn	wenn cc erfüllt, dann wie CALL nn	.	.	.	.	.	.	11 cc 100 - n - - n -	10 17	wenn cc nicht erf. wenn cc erfüllt
RET	PC_L:=(SP) PC_H:=(SP+1) SP:=SP+2	.	.	.	.	.	.	11 001 001	10	
Rcc	wenn cc erfüllt, dann wie RET	.	.	.	.	.	.	11 cc 000	5 11	wenn cc nicht erf. wenn cc erfüllt
RETI	Rücksprung von Interrupt	.	.	.	.	.	.	11 101 101 01 001 101	14	u | P 000 | 00H 001 | 08H 010 | 10H 011 | 18H 100 | 20H 101 | 28H 110 | 30H 111 | 38H
RETN	Rücksprung vom nicht maskierbaren Interrupt	.	.	.	.	.	.	11 101 101 01 000 101	14	
RST p	(SP-1):=PC_H (SP-2):=PC_L PC:=P SP:=SP-2	.	.	.	.	.	.	11 u 111	11	

Assembler-sprache	Symbolische Operation	Flags CY	Z	P/V	S	N	H	Op.-Kode 76 543 210	Takte	Bemerkungen
Ein- und Ausgabebefehle										
IN n	A:= (n)	.	.	.	.	.	.	11 011 011 - n -	11	bei IN n: n an A0-A7 A an A8-A15
IN r	r:= (C)	.	↕	P	↕	0	↕	11 101 101 01 r 000	12	sonst: C an A0-A7 B an A8-A15
INF	Flags stellen entsprechend.. (C)	↕		P	↕	0	↕	11 101 101 01 110 000	12	
INI	(HL):= (C) B:=B-1 HL:=HL+1	.	↕c	x	x	1	x	11 101 101 10 100 010	16	
INIR	(HL):=(C) B:=B-1 HL:=HL+1 Wiederholung bis B=0	.	1	x	x	1	x	11 101 101 10 110 010	21 16	wenn B≠0 wenn B=0
IND	(HL):=(C) B:=B-1 HL:=HL-1	.	↕c	x	x	1	x	11 101 101 10 101 010	16	
INDR	(HL):=(C) B:=B-1 HL:=HL-1 Wiederholung bis B=0	.	↕c	x	x	1	x	11 101 101 10 111 010	21 16	wenn B≠0 wenn B=0
OUT n	(n):=A	.	.	.	.	.	.	11 010 011 - n -	11	bei OUT n n an A0-A7 A an A8-A15
OUT r	(C):=r	.	.	.	.	.	.	11 101 101 01 r 001	12	sonst: C an A0-A7 B an A8-A15
OUTI	(C):=(HL) B:=B-1 HL:=HL+1	.	↕c	x	x	1	x	11 101 101 10 100 011	16	
OTIR	(C):=(HL) B:=B-1 HL:=HL+1 Wiederholung bis B=0	.	1	x	x	1	x	11 101 101 10 110 011	21 16	wenn B≠0 wenn B=0
OUTD	(C):=(HL) B:=B-1 HL:=HL-1	.	↕c	x	x	1	x	11 101 101 10 101 011	16	
OTDR	(C):=(HL) B:B-1 HL:=HL-1 Wiederholung bis B=0	.	1	x	x	1	x	11 101 101 10 111 011	21 16	wenn B≠0 wenn B=0

Befehlsliste Hexadez.Darstellung

Erläuterungen zur Befehlsliste

nn Bytekonstante
vv Verschiebekonstante
hh höherwertiger Teil einer 16-Bit-Konstante
ll niederwertiger Teil einer 16-Bit-Konstante

Anordnung nach alphabetischer Reihenfolge der Assemblerbefehle

8E	ADC	(HL)	A5	AND	L	CB61	BIT	4,C			
DD8Evv	ADC	(IX+d)	E6nn	AND	n	CB62	BIT	4,D			
FD8Evv	ADC	(IY+d)	CB 46	BIT	0,(HL)	CB63	BIT	4,E			
8F	ADC	A	DDCBvv46	BIT	0,(IX+d)	CB64	BIT	4,H			
88	ADC	B	FDCBvv46	BIT	0,(IY+d)	CB65	BIT	4,L			
89	ADC	C	CB47	BIT	0,A	CB6E	BIT	5,(HL)			
8A	ADC	D	CB40	BIT	0,B	DDCBvv6E	BIT	5,(IX+d)			
8B	ADC	E	CB41	BIT	0,C	FDCBvv6E	BIT	5,(IY+d)			
8C	ADC	H	CB42	BIT	0,D	CB6F	BIT	5,A			
8D	ADC	L	CB43	BIT	0,E	CB68	BIT	5,B			
CE nn	ADC	n	CB44	BIT	0,H	CB69	BIT	5,C			
ED4A	ADC	HL,BC	CB45	BIT	0,L	CB6A	BIT	5,D			
ED5A	ADC	HL,DE	CB4E	BIT	1,(HL)	CB6B	BIT	5,E			
ED6A	ADC	HL,HL	DDCBvv4E	BIT	1,(IX+d)	CB6C	BIT	5,H			
ED7A	ADC	HL,SP	FDCBvv4E	BIT	1,(IY+d)	CB6D	BIT	5,L			
86	ADD	(HL)	CB4F	BIT	1,A	CB76	BIT	6,(HL)			
DD86vv	ADD	(IX+d)	CB48	BIT	1,B	DDCBvv76	BIT	6,(IX+d)			
FD86vv	ADD	(IY+d)	CB49	BIT	1,C	FDCBvv76	BIT	6,(IY+d)			
87	ADD	A	CB4A	BIT	1,D	CB77	BIT	6,A			
80	ADD	B	CB4B	BIT	1,E	CB70	BIT	6,B			
81	ADD	C	CB4C	BIT	1,H	CB71	BIT	6,C			
82	ADD	D	CB4D	BIT	1,L	CB72	BIT	6,D			
83	ADD	E	CB56	BIT	2,(HL)	CB73	BIT	6,E			
84	ADD	H	DDCBvv56	BIT	2,(IX+d)	CB74	BIT	6,H			
85	ADD	L	FDCBvv56	BIT	2,(IY+d)	CB75	BIT	6,L			
C6nn	ADD	n	CB57	BIT	2,A	CB7E	BIT	7,(HL)			
09	ADD	HL,BC	CB50	BIT	2,B	DDCBw7E	BIT	7,(IX+d)			
19	ADD	HL,DE	CB51	BIT	2,C	FDCBvv7E	BIT	7,(IY+d)			
29	ADD	HL,HL	CB52	BIT	2,D	CB7F	BIT	7,A			
39	ADD	HL,SP	CB53	BIT	2,E	CB78	BIT	7,B			
DD09	ADD	IX,BC	CB54	BIT	2,H	CB79	BIT	7,C			
DD19	ADD	IX,DE	CB55	BIT	2,L	CB7A	BIT	7,D			
DD29	ADD	IX,IX	CB5E	BIT	3,(HL)	CB7B	BIT	7,E			
DD39	ADD	IX,SP	DDCBvv5E	BIT	3,(IX+d)	CB7C	BIT	7,H			
FD09	ADD	IY,BC	FDCBvv5E	BIT	3,(IY+d)	CB7D	BIT	7,L			
FD19	ADD	IY,DE	CB5F	BIT	3,A	DC11hh	CAC	nn			
FD29	ADD	IY,IY	CB58	BIT	3,B	CD11hh	CALL	nn			
FD39	ADD	IY,SP	CB59	BIT	3,C	FC11hh	CAM	nn			
A6	AND	(HL)	CB5A	BIT	3,D	D411hh	CANC	nn			
DDA6vv	AND	(IX+d)	CB5B	BIT	3,E	C411hh	CANZ	nn			
FDA6vv	AND	(IY+d)	CB5C	BIT	3,H	F411hh	CAP	nn			
A7	AND	A	CB5D	BIT	3,L	EC11hh	CAPE	nn			
A0	AND	B	CB66	BIT	4,(HL)	E411hh	CAPO	nn			
A1	AND	C	DDCBvv66	BIT	4,(IX+d)	CC11hh	CAZ	nn			
A2	AND	D	FDCBvv66	BIT	4,(IY+d)	3F	CCF				
A3	AND	E	CB67	BIT	4,A	BE	CMP	(HL)			
A4	AND	H	CB60	BIT	4,B	DDBEvv	CMP	(IX+d)			

FDBEvv	CMP (IY+d)
BF	CMP A
B8	CMP B
B9	CMP C
BA	CMP D
BB	CMP E
BC	CMP H
BD	CMP L
FE nn	CMP n
EDA9	CPD
EDB9	CPDR
EDA1	CPI
EDB1	CPIR
2F	CPL
27	DAA
35	DEC (HL)
DD35vv	DEC (IX+d)
FD35vv	DEC (IY+d)
3D	DEC A
Ø5	DEC B
ØB	DEC BC
ØD	DEC C
15	DEC D
1B	DEC DE
1D	DEC E
25	DEC H
2B	DEC HL
DD2B	DEC IX
FD2B	DEC IY
2D	DEC L
3B	DEC SP
F3	DI
1Øvv	DJNZ e
FB	EI
E3	EX (SP),HL
DDE3	EX (SP),IX
FDE3	EX (SP),IY
Ø8	EXAF
EB	EX DE,HL
D9	EXX
76	HALT
ED46	IMØ
ED56	IM1
ED5E	IM2
ED78	IN A
ED4Ø	IN B
ED48	IN C
ED5Ø	IN D
ED58	IN E
ED7Ø	INF
ED6Ø	IN H
ED68	IN L
DBnn	IN n

34	INC (HL)
DD34vv	INC (IX+d)
FD34vv	INC (IY+d)
3C	INC A
Ø4	INC B
Ø3	INC BC
ØC	INC C
14	INC D
13	INC DE
1C	INC E
24	INC H
23	INC HL
DD23	INC IX
FD23	INC IY
2C	INC L
33	INC SP
EDAA	IND
EDBA	INDR
EDA2	INI
EDB2	INIR
E9	JMP (HL)
DDE9	JMP (IX)
FDE9	JMP (IY)
C3llhh	JMP nn
DAllhh	JPC nn
FAllhh	JPM nn
D2llhh	JPNC nn
C2llhh	JPNZ nn
F2llhh	JPP nn
EAllhh	JPPE nn
E2llhh	JPPO nn
CAllhh	JPZ nn
38 vv	JRC e
18 vv	JR e
3Ø vv	JRNC e
2Ø vv	JRNZ e
28 vv	JRZ e
Ø2	LD (BC),A
12	LD (DE),A
77	LD (HL),A
7Ø	LD (HL),B
71	LD (HL),C
72	LD (HL),D
73	LD (HL),E
74	LD (HL),H
75	LD (HL),L
36nn	LD (HL),n
DD77vv	LD (IX+d),A
DD7Øvv	LD (IX+d),B
DD71vv	LD (IX+d),C
DD72vv	LD (IX+d),D
DD73vv	LD (IX+d),E

DD74vv	LD (IX+d),H
DD75vv	LD (IX+d),L
DD36vvnn	LD (IX+d),n
FD77vv	LD (IY+d),A
FD7Ovv	LD (IY+d),B
FD71vv	LD (IY+d),C
FD72vv	LD (IY+d),D
FD73vv	LD (IY+d),E
FD74vv	LD (IY+d),H
FD75vv	LD (IY+d),L
FD36vvnn	LD (IY+d),n
32llhh	LD (nn),A
ED43llhh	LD (nn),BC
ED53llhh	LD (nn),DE
22llhh	LD (nn),HL
DD22llhh	LD (nn),IX
FD22llhh	LD (nn),IY
ED73llhh	LD (nn),SP
ØA	LD A,(BC)
1A	LD A,(DE)
7E	LD A,(HL)
DD7Evv	LD A,(IX+d)
FD7Evv	LD A,(IY+d)
3Allhh	LD A,(nn)
7F	LD A,A
78	LD A,B
79	LD A,C
7A	LD A,D
7B	LD A,E
7C	LD A,H
ED57	LD A,I
7D	LD A,L
3Enn	LD A,n
ED5F	LD A,R
46	LD B,(HL)
DD46vv	LD B,(IX+d)
FD46vv	LD B,(IY+d)
47	LD B,A
4Ø	LD B,B
41	LD B,C
42	LD B,D
43	LD B,E
44	LD B,H
45	LD B,L
Ø6nn	LD B,n
ED4Bllhh	LD BC,(nn)
Ø1llhh	LD BC,nn
4E	LD C,(HL)
DD4Evv	LD C,(IX+d)
FD4Evv	LD C,(IY+d)
4F	LD C,A
48	LD C,B

49	LD	C,C	6F	LD	L,A	D5	PUSH	DE
4A	LD	C,D	68	LD	L,B	E5	PUSH	HL
4B	LD	C,E	69	LD	L,C	DDE5	PUSH	IX
4C	LD	C,H	6A	LD	L,D	FDE5	PUSH	IY
4D	LD	C,L	6B	LD	L,E	D8	RC	
ØEnn	LD	C,n	6C	LD	L,H	CB86	RES	Ø,(HL)
56	LD	D,(HL)	6D	LD	L,L	DDCBvv86	RES	Ø,(IX+d)
DD56vv	LD	D,(IX+d)	2Enn	LD	L,n	FDCBvv86	RES	Ø,(IY+d)
FD56vv	LD	D,(IY+d)	ED4F	LD	R,A	CB87	RES	Ø,A
57	LD	D,A	ED7Bllhh	LD	SP,(nn)	CB8Ø	RES	Ø,B
5Ø	LD	D,B	F9	LD	SP,HL	CB81	RES	Ø,C
51	LD	D,C	DDF9	LD	SP,IX	CB82	RES	Ø,D
52	LD	D,D	FDF9	LD	SP,IY	CB83	RES	Ø,E
53	LD	D,E	31llhh	LD	SP,nn	CB84	RES	Ø,H
54	LD	D,H	EDA8	LDD		CB85	RES	Ø,L
55	LD	D,L	EDB8	LDDR		CB8E	RES	1,(HL)
16nn	LD	D,n	EDAØ	LDI		DDCBvv8E	RES	1,(IX+d)
ED5BllhhLD		DE,(nn)	EDBØ	LDIR		FDCBvv8E	RES	1,(IY+d)
11llhh	LD	DE,nn	ED44	NEG		CB8F	RES	1,A
5E	LD	E,(HL)	ØØ	NOP		CB88	RES	1,B
DD5Evv	LD	E,(IX+d)	B6	OR	(HL)	CB89	RES	1,C
FD5Evv	LD	E,(IY+d)	DDB6vv	OR	(IX+d)	CB8A	RES	1,D
5F	LD	E,A	FDB6vv	OR	(IY+d)	CB8B	RES	1,E
58	LD	E,B	B7	OR	A	CB8C	RES	1,H
59	LD	E,C	BØ	OR	B	CB8D	RES	1,L
5A	LD	E,D	B1	OR	C	CB96	RES	2,(HL)
5B	LD	E,E	B2	OR	D	DDCBvv96	RES	2,(IX+d)
5C	LD	E,H	B3	OR	E	FDCBvv96	RES	2,(IY+d)
5D	LD	E,L	B4	OR	H	CB97	RES	2,A
1Enn	LD	E,n	B5	OR	L	CB9Ø	RES	2,B
66	LD	H,(HL)	F6nn	OR	n	CB91	RES	2,C
DD66vv	LD	H,(IX+d)	EDBB	OTDR		CB92	RES	2,D
FD66vv	LD	H,(IY+d)	EDB3	OTIR		CB93	RES	2,E
67	LD	H,A	ED79	OUT	A	CB94	RES	2,H
6Ø	LD	H,B	ED41	OUT	B	CB95	RES	2,L
61	LD	H,C	ED49	OUT	C	CB9E	RES	3,(HL)
62	LD	H,D	EDAB	OUTD		DDCBvv9E	RES	3,(IX+d)
63	LD	H,E	ED51	OUT	D	FDCBvv9E	RES	3,(IY+d)
64	LD	H,H	ED59	OUT	E	CB9F	RES	3,A
65	LD	H,L	ED61	OUT	H	CB98	RES	3,B
26nn	LD	H,n	EDA3	OUTI		CB99	RES	3,C
2Allhh	LD	HL,(nn)	ED69	OUT	L	CB9A	RES	3,D
21llhh	LD	HL,nn	D3nn	OUT	n	CB9B	RES	3,E
ED47	LD	I,A	F1	POP	AF	CB9C	RES	3,H
DD2AllhhLD		IX,(nn)	C1	POP	BC	CB9D	RES	3,L
DD21llhhLD		IX,nn	D1	POP	DE	CBA6	RES	4,(HL)
FD2AllhhLD		IY,(nn)	E1	POP	HL	DDCBvvA6	RES	4,(IX+d)
FD21llhhLD		IY,nn	DDE1	POP	IX	FDCBvvA6	RES	4,(IY+d)
6E	LD	L,(HL)	FDE1	POP	IY	CBA7	RES	4,A
DD6Evv	LD	L,(IX+d)	F5	PUSH	AF	CBAØ	RES	4,B
FD6Evv	LD	L,(IY+d)	C5	PUSH	BC	CBA1	RES	4,C

CBA2	RES	4,D
CBA3	RES	4,E
CBA4	RES	4,H
CBA5	RES	4,L
CBAE	RES	5,(HL)
DDCBvvAE	RES	5,(IX+d)
FDCBvvAE	RES	5,(IY+d)
CBAF	RES	5,A
CBA8	RES	5,B
CBA9	RES	5,C
CBAA	RES	5,D
CBAB	RES	5,E
CBAC	RES	5,H
CBAD	RES	5,L
CBB6	RES	6,(HL)
DDCBvvB6	RES	6,(IX+d)
FDCBvvB6	RES	6,(IY+d)
CBB7	RES	6,A
CBBØ	RES	6,B
CBB1	RES	6,C
CBB2	RES	6,D
CBB3	RES	6,E
CBB4	RES	6,H
CBB5	RES	6,L
CBBE	RES	7,(HL)
DDCBvvBE	RES	7,(IX+d)
FDCBvvBE	RES	7,(IY+d)
CBBF	RES	7,A
CBB8	RES	7,B
CBB9	RES	7,C
CBBA	RES	7,D
CBBB	RES	7,E
CBBC	RES	7,H
CBBD	RES	7,L
C9	RET	
ED4D	RETI	
ED45	RETN	
CB16	RL	(HL)
DDCBvv16	RL	(IX+d)
FDCBvv16	RL	(IY+d)
17	RLA	
CB17	RL	A
CB1Ø	RL	B
CB11	RL	C
CB12	RL	D
CB13	RL	E
CB14	RL	H
CB15	RL	L
CBØ6	RLC	(HL)
DDCBvvØ6	RLC	(IX+d)
FDCBvvØ6	RLC	(IY+d)
Ø7	RLCA	

CBØ7	RLC	A
CBØØ	RLC	B
CBØ1	RLC	C
CBØ2	RLC	D
CBØ3	RLC	E
CBØ4	RLC	4
CBØ5	RLC	L
ED6F	RLD	
F8	RM	
DØ	RNC	
CØ	RNZ	
FØ	RP	
E8	RPE	
EØ	RPO	
CB1E	RR	(HL)
DDCBvv1E	RR	(IX+d)
FDCBvv1E	RR	(IY+d)
1F	RRA	
CB1F	RR	A
CB18	RR	B
CB19	RR	C
CB1A	RR	D
CB1B	RR	E
CB1C	RR	H
CB1D	RR	L
CBØE	RRC	(HL)
DDCBvvØE	RRC	(IX+d)
FDCBvvØE	RRC	(IY+d)
ØF	RRCA	
CBØF	RRC	A
CBØ8	RRC	B
CBØ9	RRC	C
CBØA	RRC	D
CBØB	RRC	E
CBØC	RRC	H
CBØD	RRC	L
ED67	RRD	
C7	RST	ØØH
CF	RST	Ø8H
D7	RST	1ØH
DF	RST	18H
E7	RST	2ØH
EF	RST	28H
F7	RST	3ØH
FF	RST	38H
C8	RZ	
9E	SBC	(HL)
DD9Evv	SBC	(IX+d)
FD9Evv	SBC	(IY+d)
9F	SBC	A
98	SBC	B
99	SBC	C
9A	SBC	D

9B	SBC	E
9C	SBC	H
ED42	SBC	HL,BC
ED52	SBC	HL,DE
ED62	SBC	HL,HL
ED72	SBC	HL,SP
9D	SBC	L
DEnn	SBC	n
37	SCF	
CBC6	SET	Ø,(HL)
DDCBvvC6	SET	Ø,(IX+d)
FDCBvvC6	SET	Ø,(IY+d)
CBC7	SET	Ø,A
CBCØ	SET	Ø,B
CBC1	SET	Ø,C
CBC2	SET	Ø,D
CBC3	SET	Ø,E
CBC4	SET	Ø,H
CBC5	SET	Ø,L
CBCE	SET	1,(HL)
DDCBvvCE	SET	1,(IX+d)
FDCBvvCE	SET	1,(IY+d)
CBCF	SET	1,A
CBC8	SET	1,B
CBC9	SET	1,C
CBCA	SET	1,D
CBCB	SET	1,E
CBCC	SET	1,H
CBCD	SET	1,L
CBD6	SET	2,(HL)
DDCBvvD6	SET	2,(IX+d)
FDCBvvD6	SET	2,(IY+d)
CBD7	SET	2,A
CBDØ	SET	2,B
CBD1	SET	2,C
CBO2	SET	2,D
CBO3	SET	2,E
CBD4	SET	2,H
CBD5	SET	2,L
CBDE	SET	3,(HL)
DDCBvvDE	SET	3,(IX+d)
FDCBvvDE	SET	3,(IY+d)
CBDF	SET	3,A
CBD8	SET	3,B
CBD9	SET	3,C
CBDA	SET	3,D
CBDB	SET	3,E
CBDC	SET	3,H
CBDD	SET	3,L
CBE6	SET	4,(HL)
DDCBvvE6	SET	4,(IX+d)
FDCBvvE6	SET	4,(IY+d)

CBE7	SET	4,A
CBEØ	SET	4,B
CBE1	SET	4,C
CBE2	SET	4,D
CBE3	SET	4,E
CBE4	SET	4,H
CBE5	SET	4,L
CBEE	SET	5,(HL)
DDCBvvEE	SET	5,(IX+d)
FDCBvvEE	SET	5,(IY+d)
CBEF	SET	5,A
CBE8	SET	5,B
CBE9	SET	5,C
CBEA	SET	5,D
CBEB	SET	5,E
CBEC	SET	5,H
CBED	SET	5,L
CBF6	SET	6,(HL)
DDCBvvF6	SET	6,(IX+d)
FDCBvvF6	SET	6,(IY+d)
CBF7	SET	6,A
CBFØ	SET	6,B
CBF1	SET	6,C
CBF2	SET	6,D
CBF3	SET	6,E
CBF4	SET	6,H
CBF5	SET	6,L
CBFE	SET	7,(HL)
DDCBvvFE	SET	7,(IX+d)
FDCBvvFE	SET	7,(IY+d)
CBFF	SET	7,A
CBF8	SET	7,B
CBF9	SET	7,C
CBFA	SET	7,D
CBFB	SET	7,E
CBFC	SET	7,H
CBFD	SET	7,L
CB26	SLA	(HL)
DDCBvv26	SLA	(IX+d)
FDCBvv26	SLA	(IY+d)
CB27	SLA	A
CB2Ø	SLA	B
CB21	SLA	C
CB22	SLA	D
CB23	SLA	E
CB24	SLA	H
CB25	SLA	L
CB2E	SRA	(HL)
DDCBvv2E	SRA	(IX+d)
FDCBvv2E	SRA	(IY+d)
CB2F	SRA	A
CB28	SRA	B
CB29	SRA	C
CB2A	SRA	D
CB2B	SRA	E
CB2C	SRA	H
CB2D	SRA	L
CB3E	SRL	(HL)
DDCBvv3E	SRL	(IX+d)
FDCBvv3E	SRL	(IY+d)
CB3F	SRL	A
CB38	SRL	B
CB39	SRL	C
CB3A	SRL	D
CB3B	SRL	E
CB3C	SRL	H
CB3D	SRL	L
96	SUB	(HL)
DD96vv	SUB	(IX+d)
FD96vv	SUB	(IY+d)
97	SUB	A
9Ø	SUB	B
91	SUB	C
92	SUB	D
93	SUB	E
94	SUB	H
95	SUB	L
D6nn	SUB	n
AE	XOR	(HL)
DDAEvv	XOR	(IX+d)
FDAEvv	XOR	(IY+d)
AF	XOR	A
A8	XOR	B
A9	XOR	C
AA	XOR	D
AB	XOR	E
AC	XOR	H
AD	XOR	L
EEnn	XOR	n